Exploring AI for Auto-tuning through Sparse Matrix Image Information

Takahiro Katagiri (Nagoya University, Japan)

Collaboratos:
Mr. **Shota Aoki** (ex-M2, Graduate School of Informatics),
Prof. **Masatoshi Kawai** (ITC, Nagoya University)

45th ASE Seminar (Advanced Supercomputing Environment): International Workshop on “Integration of Simulation/Data/Learning and Beyond” 13:45 - 14:00, November 29 (Wednesday), 2023
東京大学柏Ⅱキャンパス 情報基盤センター 4階 T412, 柏Ⅱキャンパス 情報基盤センター

名古屋大学
NAGOYA UNIVERSITY

Aim of the Group

1. Development of AT framework for mixed-precision computations and power consumption optimization
   - Directive-based AT language:
     Proposal of a new extension functions for ppOpen-AT
2. Development of AT facility with AI
   - Adaptation of deep learning technology for auto-tuning facility
   - Adaptation of Explainable AI (XAI)

Outline

- Background
- PICCG Parameter Tuning
- Conclusion

Outline

- Background
- PICCG Parameter Tuning
- Conclusion

Background

- Utilizing unverified AI prediction results contributes to various social problems.
  - Human sense verification is essential for ensuring the accuracy of AI predictions.
- To minimize production costs, there is extensive research in the field of Software Auto-tuning (AT) technology.
  - The integration of AI into AT facilities is currently advancing.

→ It is crucial to validate the “explainability of AI” by tuning performance parameters on numerical libraries.

→ Scientific XAI (SXAI)

Explainable AI (XAI)

- Can we explain result from machine learning?
  - **Explainable AI (XAI)**
- This is sorted as the following two categories [1]
  - **Explainability**
    - AI technology for reason of prediction to be understood by humans easily. Ex) LIME, SHAP
  - **Interpretability**
    - AI technology to show process for prediction by analysing inner structures. Ex) Make a decision tree.

→In this study, we focus on the “Explainability.”

[1] Otsubo et. al, “XAI (Explainable AI) ：What does AI think in that time? ”, RIC telecom, 2021, In Japanese

Global Explanations and Local Explanations

- **Global Explanations**
  - To understand “whole attribution of AI model. ”[1]
  - Ex) SHAP
- **Local Explanations**
  - To understand “Judgement result of prediction for each result.” [1]
  - Ex) LIME

→ In this research, the both of all are treated for AI of AT to numerical libraries.

[1] Otsubo et. al, “XAI (Explainable AI) ：What does AI think in that time? ”, RIC telecom, 2021, In Japanese

LIME (Local Interpretable Model-agnostic explanations) [2]

- **Local Surrogate Model**
  - An explainable model for explanation of each prediction on Blackbox model.
- Show reason to be understood by humans.
- Explain prediction from classifier to analyze contributions for each factor.
  - Obtaining AI model by varied nearest data from target explainable data.
- Adaptable for arbitrary clarifiers.
- Drawbacks
  1. Explanation is not stable. (By using random factors.)
  2. Needs tuning for hyper parameters.
  3. There is a case that cannot explain.

     → The other approaches are required.

[2] M. T. Ribeiro, S. Singh, and C. Guestrin: Why should I trust you?: Explaining the predictions of any classifier, Proc. of 22nd ACM SIGKDD, pp.1135-1144, 2016.

SHAP（SHapley Additive exPlanations）[3]

- Shapley Value, by theory of cooperative game, is adapted to machine learning.
  - There is a reasonable background in viewpoint of theory.
- Approximate Shapley value is calculated.
  - Tree-based ensemble models : High speed and accurate Shapley value.
  - Deep learning models : High speed and approximate Shapley value.
  - General Algorithms : Estimated Shapley value.
- Drawback
  1. Computational complexity is high.
     → Utilize approximate value in usual.

[3] S. Lundberg, S-I. Lee, A Unified Approach to Interpreting Model Predictions, 2017
https://arxiv.org/abs/1705.07874

Outline

- Background
- PICCG Parameter Tuning
- Conclusion

PICCG Parameter Tuning

名古屋大学
NAGOYA UNIVERSITY

Preconditioner of Incomplete Cholesky Factorization

- Incomplete Cholesky (IC) factorization preprocessing is an iterative method for solving a system of linear equations, such as $Ax = b (A \in \mathbb{R}^{n \times n},\ x,\ b \in \mathbb{R}^n)$, with a sparse symmetric matrix A as coefficients.
- IC is used as a preconditioner for conjugate gradient (CG) method. The algorithm is called *PICCG method*.
- A zero-valued element of A may become non-zero in the decomposition matrix U. This is called *fill-in*.
- The basic IC decomposition rejects (treats as 0) all fill-ins. It can keep the number of nonzero elements small.
- The basic IC is also reduce the computational complexity, but if A and $U^t DU$ are very different, then it may not work as a preconditioner matrix.

IC Preconditioner with a Threshold

- Set the maximum fill-in level (m) and the threshold value (t).
  - Treat fill-in smaller than the threshold value as 0 for fill-in below the maximum fill-in level, and generate fill-in above the threshold value.

$$d_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} u_{i,k} d_{i,i} u_{k,j}$$

$$f_{i,j} = \begin{cases} 0, & a_{i,j} \neq 0 \\ f_{i,k} + f_{k,i} + 1, & else \end{cases}$$

$$u_{i,j} = \begin{cases} d_{i,j}^{-1}(a_{i,j} - \sum_{k=1}^{i-1} u_{i,k} d_{i,i} u_{k,j}), & f_{i,j} \leq m \wedge |u_{i,j}| \geq t \\ 0, & else \end{cases}$$

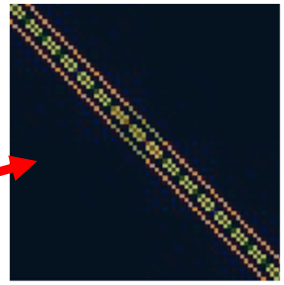
IC Preconditioner with a Threshold for $A \approx U^t D U,\ A, U, D \in \mathbb{R}^{n \times n}$

$a_{i,j}$:i,j element of$A$, $d_{i,i}$:i,i element of D, $u_{i,j}$: i,j element of$U$, $f_{i,j}$: fill-in level of$u_{i,j}$,
t: a threshold treating 0, m: maximum fill-in level

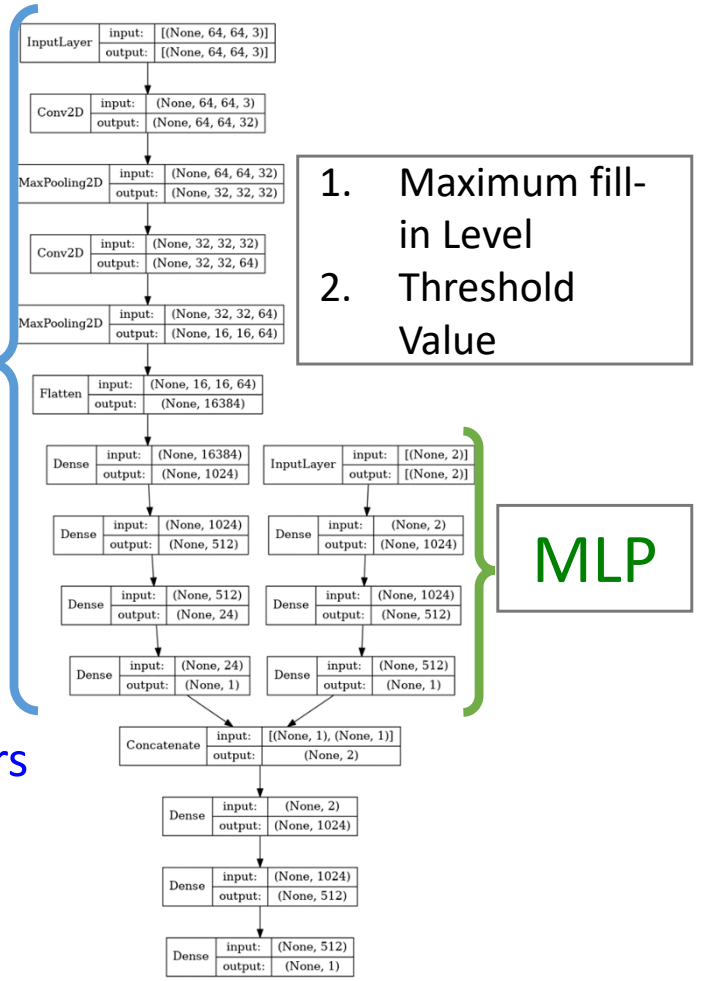
Parameters affecting execution time

Regression Model to Predict Calculation Time for ICCG Method with Threshold

- Utilize Tensorflow
- Input
  1. Feature image of coefficient matrix (created by the method of Yamada et al. [4])
  2. Maximum fill-in level and Threshold
- Output : Execution time of ICCG method with threshold



CNN



1. Maximum fill-in Level
2. Threshold Value

MLP

convolutional layer
+
Pooling layer: 2 layers
Fully connected layers: 3 layers

Learning Settings	
Number of Epochs	200
Batch Size	256
Activation Function	ReLU
Optimization	Adam Method
Loss Function	mean squared error

[4] K. Yamada, T. Katagiri, H. Takizawa, K. Minami, M. Yokokawa, T. Nagai, M. Ogino, ”Preconditioner Auto-Tuning Using Deep Learning for Sparse Iterative Algorithms”, 2018 Sixth International Symposium on Comput. and Networking Workshops(CANDARW), pp. 257-262 (2018).

Making Dataset (Training and Test Data)

- Measure execution time under the following conditions with the Supercomputer “Flow" Type I subsystem:
  - Coefficient Matrix
    - Sizes : 4096x4096, 32768x32768, 262144x262144 : 3 Kinds
    - Different of Condition Numbers for each size: 90 Kinds
  - Maximum fill-in Level
    - 0, 1, 2 : 3 kinds
  - Threshold Values
    - 0.001 ~ 0.02 by stridden 0.001: 199 Kinds
- Training data for each size: **41,073 Kinds**
- Testing data for each size: **10,269 Kinds**
- a regression model to predict execution time of PICCG method with threshold for each size is generated.

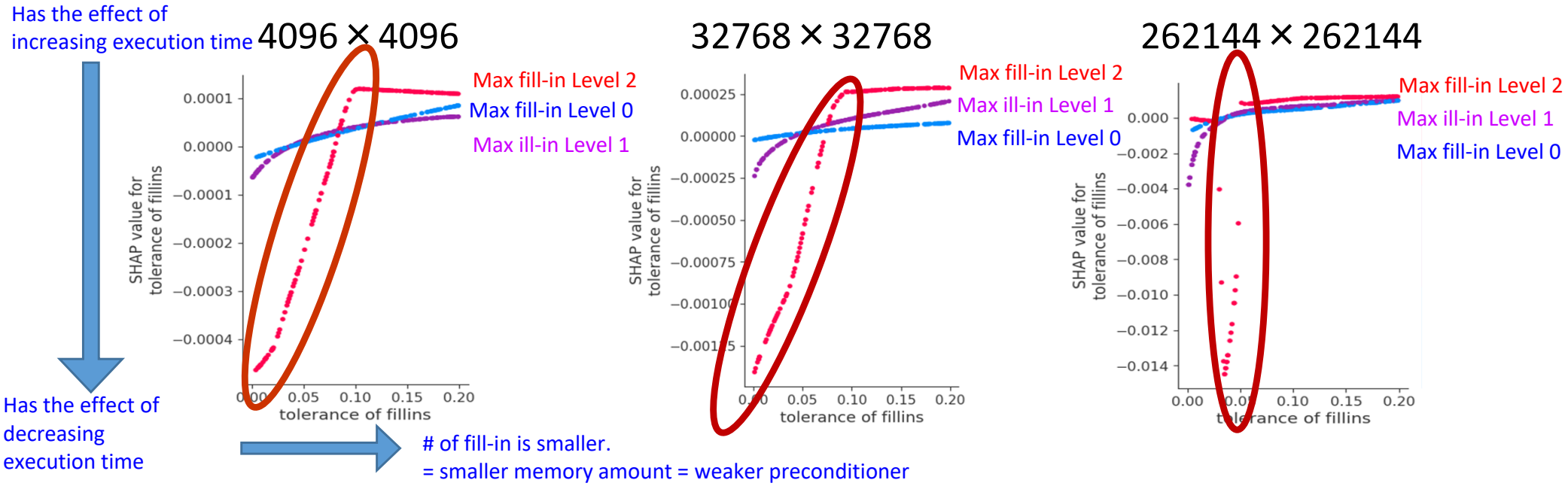
Result of Machine Learning

- Minimum, average, and maximum values of measured values for all data.
- Mean absolute error between the observed execution time and the model's prediction time on the test data

Problem sizes	Estimated Min. Time (s)	Average Time (s)	Estimated Max. Time (s)	Mean Absolute Error (ratio to Ave. Time)
4096 x 4096	0.00528	0.00746	0.0423	0.000888 (11.9%)
32768 x 32768	0.0140	0.0196	0.0379	0.000659 (**3.3%**)
262144 x262144	0.0548	0.0980	0.118	0.00409 (**4.1%**)

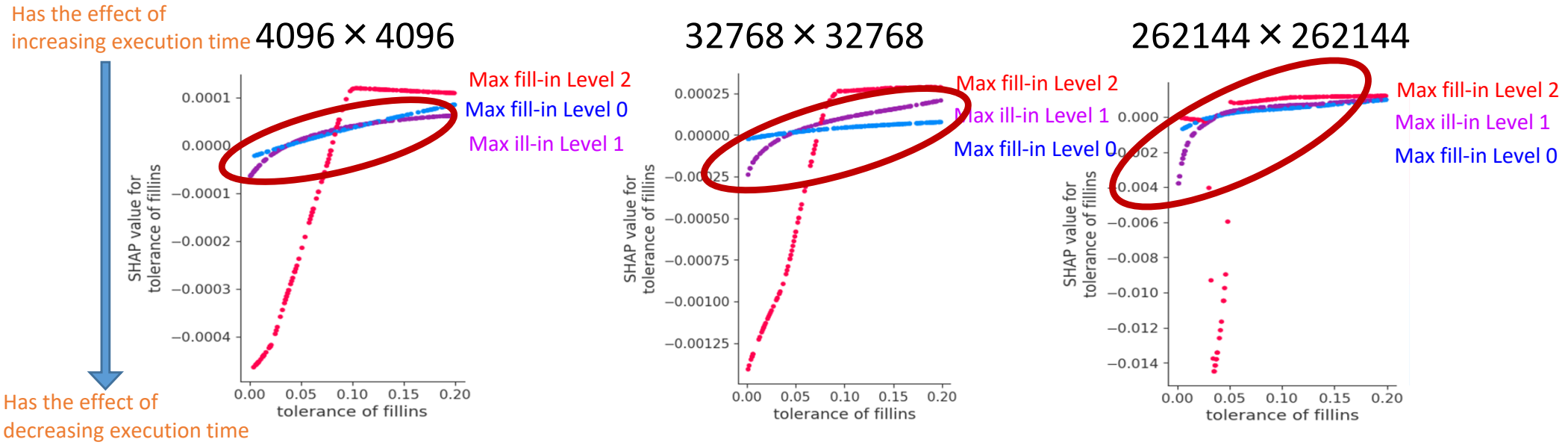
- There is a mean absolute error of 10% or less with respect to the average value, except for 4096x4095. Hence, the model is reasonable.
- We show SHAP's explanation of how the maximum fill-in level and threshold affect the computation time for this model in next slide.

Description by SHAP (In case of max fill-in level 2)



- When the maximum fill-in level is 2 (red point)
  - For 4096 x 4096 and 32768 x 32768 matrices, according to smaller the threshold, shorter execution time is observed. → Reasonable
  - In the 26,2144 x 26,2144 matrix, the execution time tends to be shorten when the threshold is between 0.035 and 0.050.
    → Reasonable (in the tanning data)

Description by SHAP (In case of max fill-in level 0 and 1)



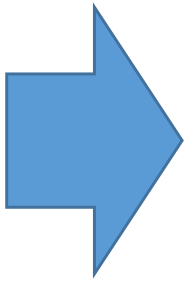
In case of maximum fill-in level is 0 (blue point) and 1 (purple point)

- Actual: Since there is no fill-in at less than fill-in level 1 in this training data, **the execution time does not depend on the threshold.** →The behavior should not change whether the maximum fill-in level is 0 or 1.
- Explanation of SHAP: according to smaller the threshold, the shorter the execution time is observed. → Not reasonable
- Explanation of SHAP: according to higher the maximum fill-in level, more susceptible to the threshold is observed. → Not reasonable

Adapting One-Hot encoding

- One-Hot encoring: A method of converting qualitative data into variables (dummy variables) that express 0 and 1.

	Feature Value
Max fill-in Level 0	0
Max fill-in Level 1	1
Max fill-in Level 2	2



	Feature Value 0	Feature Value 1	Feature Value 2
Max fill-in Level 0	1	0	0
Max fill-in Level 1	0	1	0
Max fill-in Level 2	0	0	1

- The maximum fill-in level is represented by three features.
- The corresponding feature quantity is set to 1, and the others are set to 0.
- One-hot encoding is adapted for the same training data, then learn data without changing the conditions for “the number of features in the numerical data.”

Comparison with Machine Learning Result

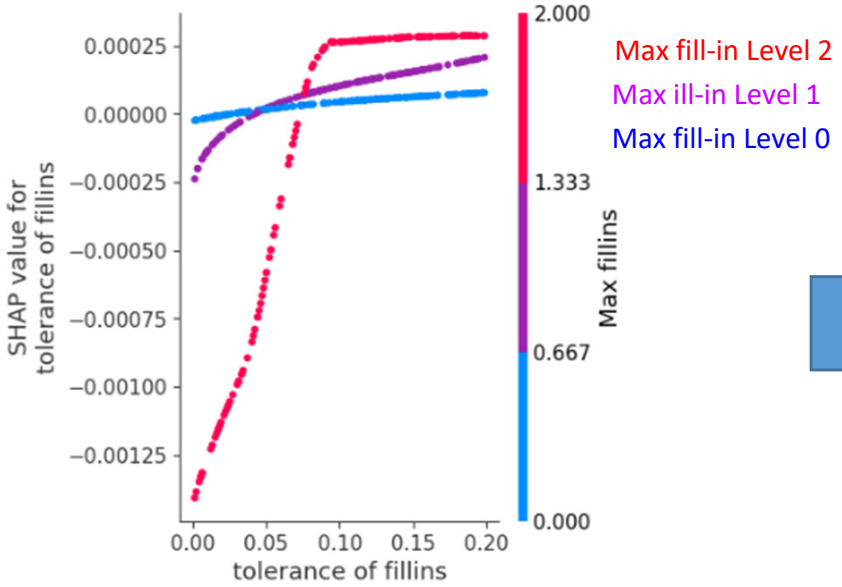
Problem Sizes	Mean Absolute Error without One-Hot Encoding	Mean Absolute Error without One-Hot Encoding (Improvement %)
4096 x 4096	0.000888	0.000204 (435%)
32768 x 32768	0.000659	0.000526 (125%)
262144 x 262144	0.00409	0.00370 (110%)

For all problem sizes, the mean absolute error decreased by One-Hot encoding.

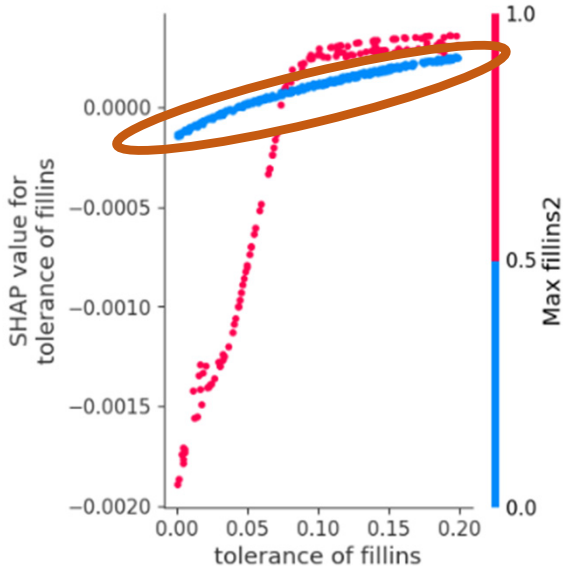
→ The model performance is improved.

Comparison of descriptions by SHAP

32768 x 32768 without One-Hot Cording



32768 x 32768 with One-Hot cording



The erroneous explanation that “the larger the maximum fill-in level is, the more likely it is to be affected by the 0 threshold” has been removed.

→ Closer to human interpretation.

Outline

- Background
- PICCG Parameter Tuning
- Conclusion

Conclusion Remarks

Explainable AI (XAI) tools can be adapted to AI outputs in the context of anomaly detection. Even within a sparse iterative algorithm, they can provide coherent explanations.

- Utilizing XAI tools such as LIME and SHAP, we derive sensible explanations based on the distinctive features of the targets.
- We have identified a pivotal traditional technique for refining AI models: the implementation of One-Hot encoding.

Future work

1. Development of AT method with XAI.
   - Reinforcement learning for explainable variables with **strong factors**.
   - Reduce AT time by reducing explainable variables with **weak factors**.
2. Adaptation of AT for mixed-precision computations.
   - Automatic selection of double and single computations in PICCG method.